This is a list of all substantial corrections made to *Computers & Typesetting* since the beginning of 2014. (More precisely, it lists errors corrected since the 19th printing of Volume A, the 9th printing of Volume B, the 8th printing of Volume C, the 6th printing of Volume D, and the 7th printing of Volume E. But it omits changes that are “purely cosmetic.”) Corrections made to the softcover version of *The TeXbook*, beginning with its 32nd printing, are the same as corrections to Volume A. Corrections to the softcover version of *The METAFONTbook*, beginning with its 11th printing, are the same as corrections to Volume C. Changes to the mini-indexes and master indexes of Volumes B, D, and E are not shown here unless they are not obviously derivable from what has been shown. Some (or all) of these errors have been corrected in the most recent printings.

---

**Page A34, line 3 from the bottom**  
(01/09/20)

not, you can say ‘\errorcontextlines=100 \oops’ and try again. (That will usually

**Page A43, line 6**  
(07/24/14)

keyboard, or that have been preempted for formatting?

**Page A49, cummings quote**  
(08/03/19)

(delete the period at the end of the line)

**Page A66, line 3 from the bottom**  
(08/26/17)

Such displays of box contents will be discussed further in Chapters 12 and 27.

**Page A105, lines 9–16**  
(01/16/21)

If you say `\vadjust{(vertical mode material)}` within a paragraph, \TeX will

use internal vertical mode to insert the specified material into the vertical list

that encloses the paragraph, immediately after whatever line contained the position of the \vadjust. For example, you can say `\vadjust{\kern1pt}` to increase the amount

of space between lines of a paragraph if those lines would otherwise come out too close
together. (The author did that in the current line, just to illustrate what happens.)

Also, if you want to make sure that a page break will occur immediately after a certain

line, you can say `\vadjust{\eject}` anywhere in that line.

**Page A122, lines 3–8**  
(11/24/19)

`\count255, \dimen255, \skip255, \muskip255,` and `\toks255` are traditionally kept

available for such purposes. Furthermore, plain \TeX reserves `\dimen0` to `\dimen9`,

`\skip0` to `\skip9`, `\muskip0` to `\muskip9`, and `\box0` to `\box9` for “scratchwork”; these

registers are never allocated by the \new... operations. We have seen that `\count0` through `\count9` are special, and `\box255` also turns out to be special; so those registers

should be avoided unless you know what you are doing.
Individual symbols; \texttt{\textbackslash left...\right} constructions are treated as “inner” subformulas, which means that they will be surrounded by additional space in certain circumstances. All other subformulas are generally treated as ordinary symbols, whether they are formed by \texttt{\overline} or \texttt{\hbox} or \texttt{\vcenter} or by simply being enclosed in braces. Thus, \texttt{\mathord} isn’t really a necessary part of the \TeX{} language; instead of typing \texttt{$1\mathord,234$} you can get the same effect from \texttt{'$1(,234$'}.  

Inner is an inner atom produced by \texttt{\textbackslash left...\right}; subformulas delimited by \texttt{\textbackslash left} and \texttt{\textbackslash right} are treated as type Inner. The following table is used to determine the spacing between pairs of adjacent atoms:  

Just after a token such as \texttt{$3$} that begins math mode, to see if another token of category 3 follows.  

\begin{itemize}
  \item The \texttt{\vsplit} operation is also explained in Chapter 15. In math modes an additional type of box is available: \texttt{\vcenter\{vertical mode material\}} (see Chapter 17).  
\end{itemize}

Only two tabs are set in this case, because only two \&’s appear in the sample line. (A sample line usually ends with \&\texttt{\cr}, as it does here, because text material between the last tab and \texttt{\cr} isn’t used for anything.)
Page A252, lines 5–7 (12/25/20)
blank, and the footline is normally a centered page number, but you can specify any headline and footline that you want by changing the token lists `\headline` and `\footline`. For example,

Page A253, lines 7–9 from the bottom (10/27/20)

\everypar or \errhelp, except that \TeX{} retains the begin-group symbol ‘{' at the beginning and the end-group symbol ‘}’ at the end. These grouping characters help to keep the output routine from interfering with what \TeX{} was doing

Page A256, line 19 (08/28/15)
\baselineskip=24pt \lineskip=0pt

Page A277, lines 9 and 10 from the bottom (08/26/17)

⟨hyphenation assignment⟩−→ \hyphenation{filler}{(hyphenations)}
| \patterns{filler}{(patterns)}

Page A286, bottom two lines (and affecting the top lines of page 287) (08/26/17)
stands for zero or more ⟨assignment⟩ commands other than \setbox, possibly with ⟨filler⟩. If the assignments are not followed by a ⟨character⟩, where ⟨character⟩ stands

Page A287, lines 11–17 (04/22/20)
\discretionary{disc text}{disc text}{disc text}. This command has the same effect as in horizontal mode (see Chapter 25), but the third ⟨disc text⟩ must produce an empty list.

Page A292, lines 8–10 (04/22/20)
\discretionary{disc text}{disc text}{disc text}. A ⟨disc text⟩ has the form ‘⟨filler⟩⟨horizontal mode material⟩’, where the material is processed in restricted horizontal mode and should contain only fixed-width things. More precisely, the horizontal list formed by each ⟨disc text⟩ must consist only of characters, ligatures, kerns, boxes, and rules; there should be no glue or penalty items, etc. This command appends a discretionary item to the current list; see Chapter 14 for the meaning of a discretionary item. The space factor is not changed.

Page A299, line 11 from the bottom (11/01/20)
is corrupted or was prepared for a different version of \TeX{}.

Page A305, bottom line (06/30/20)
\setbox0=\hbox{#1}\advance\dimen0 by -\wd0 }.
Page A309, line 2 becomes two lines (12/06/20)
represent text entered from the user’s terminal, or with ‘<insert>’, when they represent
text inserted during error recovery).

Page A316, lines 17 and 18 from the bottom (09/03/15)
(The next line must also not be too tall.) Here \specialstar is a box of height zero
and depth \strutdepth, and it puts an asterisk in the left margin:

Page A320, lines 5–9 from the bottom (06/27/15)
17.21. Assigning \delcode’{ would not work to allow ‘\left’; because the brace
has category 1 and isn’t a legal ⟨delim⟩. Allowing brace delimiters would be a bad
idea because it would mess up other constructions, such as arguments to macros, and
components of alignments. Moreover, a user who gets away with ‘\left’ is likely to
try also ‘\bigl’, which fails miserably.

Page A326, line 12 (08/26/17)
its natural width. The \hbox version also invokes \everyhbox and \everymath.

Page A329, line 3 of answer 20.7 (05/15/19)
the three tokens !1, #2, [1; the (replacement text) consists of the six tokens \{1, #6,
Page A329, line 6 of answer 20.7 (05/15/19)
is otherwise irrelevant. Thus, ‘\def\!!1#2[#1][##]!!#2’ would produce an essentially
Page A329, line 5 from the bottom of answer 20.7 (05/15/19)
!1<-x
Page A329, bottom line of answer 20.7 (05/15/19)
final parameter in the parameter text; ‘!1’ would have been rendered ‘#1’.

Page A332, lines 13 and 14 (08/26/17)
21.10. If you say ‘\let\the=0\edef\next {\write\cont{(token list)}}\next’; the
\write will be executed after \edef expands everything except \the.

Page A332, bottom line (11/15/19)
\+&{\bf end};\cr % note that the semicolon isn’t bold

Page A342, lines 12 and 13 (08/14/20)
of plain TeX format; but some of them are primitive (built in), such as ‘\par’ (end
of paragraph), ‘\noindent’ (beginning of non-indented paragraph), and ‘/’ (italic
Braces are used for grouping, when supplying arguments to macros; so they cannot also be used as math delimiters, or as arguments to macros such as \textbackslash big. (One could change their catcodes to 12, and use some other pair of characters for grouping; but that would not be plain \TeX.)

(2) The registers \textbackslash count255, \textbackslash dimen255, \textbackslash skip255, \textbackslash toks255, and \textbackslash muskip255 are freely available in the same way. (3) All assignments to the scratch registers whose numbers are 1, 3, 5, 7, and 9 should be \textbackslash global; all assignments to the other scratch registers (0, 2, 4, 6, 8, 255) should be non-\textbackslash global. (This prevents the phenomenon of “save stack buildup” discussed in Chapter 27.) (4) Furthermore, it’s possible to use any register in a group, if you ensure that \TeX’s grouping mechanism will restore the register when you’re done with the group, and if you are certain that other macros will not make global assignments to that register when you need it. (5) But when a register is used by several macros, or over long spans of time, it should be allocated by \textbackslash newcount, \textbackslash newdimen, \textbackslash newbox, etc. (6) Similar remarks apply to input/output streams used by \textbackslash read and \textbackslash write, to math families used by \textbackslash fam, to sets of hyphenation rules used by \textbackslash language, and to insertions (which require \textbackslash box, \textbackslash count, \textbackslash dimen, and \textbackslash skip registers all having the same number).

\def\wlog{\immediate\write-1 } % this will write on log file (only)
\outer\def\newmuskip{\alloc@3\muskip\muskipdef\@cclv}
\outer\def\newtoks{\alloc@5\toks\toksdef\@cclv}

format; it shouldn’t cost much for people to acquire all the fonts of plain \TeX in addition to the ones that they really want. Second, it is desirable on many computer systems to close as possible to the ASCII conventions. (b) Make sure that codes ‘041–’046, ‘060–’071, ‘136, ‘141–’146, and ‘160–’171 are present and that each unrepresentable in-
and \if...\fi tests, as well as special operations like \the and \input, while the latter category includes the primitive commands listed in Chapters 24–26. The expansion of $\text{\generaldisplay}$ to be invoked, with \eq defined to be $\alpha$. Furthermore, when an equation number $\beta$ is present, it should be stored in \eqn, and the test \ifeqno should be true. In such cases \ifeqno should distinguish \leqno from \eqno. Here \setbox2=\lastbox \setbox\footins=\vbox{\box2} since \lastbox will be the result of \rigidbalance, which is an hbox. \interlinepenalty5000\def\par{\endgraf\penalty5000 }} The computer file texbook.tex that generated The \TeXbook begins with a \def\bull{\vrule height.9ex width.8ex depth-.1ex \relax} % square bullet \vrule height6pt depth2pt width0pt \relax} % a strut for \insert\margin 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19. 15e. Enclose the vbox that was constructed in Rule 15c or 15d by delimiters $(\lambda, \rho)$ whose height plus depth is at least $\sigma_20$, if $C > T$, and at least $\sigma_21$ otherwise. Shift the delimiters up or down so that they are vertically centered with respect to the axis. Replace the generalized fraction by an Ord atom whose nucleus is the resulting sequence of three boxes $(\lambda, \text{vbox}, \rho)$. Go to rule 19.
of the process; the trial word consists of all the letters found in admissible items, up to a maximum of 63. Notice that all of these letters are in font \( f \).

Page A458 and following, selected amendments to the index (01/18/21)

\[ \text{[1] (progress report), 23, 119.} \]
\[ \text{\&a ( à ), 52, 356.} \]
\[ \text{\&AA ( À ), 52, 356.} \]
\[ \text{(disc text), 287, 292.} \]
\[ \text{(general text), 276, 279, 280.} \]
\[ \text{(horizontal mode material), 278, 285, 287.} \]
\[ \text{integral signs, see \textbackslash int, \textbackslashoint, \textbackslashsmallint.} \]
\[ \text{(math mode material), 287, 289–293.} \]
\[ \text{\textbackslashnull, 311, 312, 316, 332, 335, 351, 354, 360–362, 419.} \]
\[ \text{\&o ( ø ), 52, 356.} \]
\[ \text{\&O ( Ø ), 52, 356.} \]
\[ \text{programs, for computers, 38, 165, 234.} \]
\[ \text{repeating templates, see periodic preambles.} \]
\[ \text{replacement text, 200–204, 212, 280, 300, 329.} \]
\[ \text{right delimiters, see closings.} \]
\[ \text{struts, 82, 125, 131, 142, 155, 176, 245–247, 255, 329, 416, 422, 423.} \]
\[ \text{(vertical mode material), 278, 280–282, 290.} \]

Page Bv (formerly Bvii), bottom two lines (01/15/21)

all of those changes. I now believe that the final bug was discovered on 22 October 2020 and removed in version 3.141592653. The finder’s fee has converged to $327.68.

Page B2, line 10 from the bottom (01/15/21)

\texttt{define \texttt{banner} \equiv \texttt{This is \texttt{TeX}, Version 3.141592653}} \{ printed when \TeX\ starts \}

Page B4, line 8 of §7 (04/02/17)

diagnostic information for \texttt{\textbackslash tracingsparagraphs}, \texttt{\textbackslash tracingspages}, and \texttt{\textbackslash tracingsrestores}.

Page B21, lines 33 and 34 (04/02/17)

\[ ['41 \rightarrow '46, '60 \rightarrow '71, '136, '141 \rightarrow '146, '160 \rightarrow '171] \text{ must be printable. Thus, at least 80 printable characters are needed.} \]

Page B28, lines 3 and 4 (04/02/17)

not serious since we assume that this part of the program is system dependent.

Page B28, line 2 from the bottom (04/02/17)

\texttt{\texttt{var} \texttt{k: 0..23;}} \{ index to current digit; we assume that \( |n| < 10^{23} \) \}
Page B35, line 2 of §83 becomes two lines (06/27/20)

\begin{verbatim}
loop begin continue: if interaction ≠ error_stop_mode then return;
    clear_for_error_prompt; prompt_input("?\n");
\end{verbatim}

Page B36, line 11 of §84 (07/03/20)

"E": if base_ptr > 0 then if input_stack[base_ptr].name_field ≥ 256 then

Page B36, line 5 of §85 becomes two lines (07/03/20)

if base_ptr > 0 then
    if input_stack[base_ptr].name_field ≥ 256 then print("E\n
Page B40, line 5 from the bottom (08/07/20)

("Try\n
Page B58, lines 2 and 3 of §136 (10/11/20)

the values corresponding to `\hbox{}`. The sub_type field is set to min_quarterword, for historic reasons that are no longer relevant.

Page B88, line 16 (10/22/20)

The mode is temporarily set to zero while processing \write texts.

Page B102, lines 3 and following of §241 (12/11/20)

information, something special is needed. The program here simply assumes that suitable values appear in the global variables sys_time, sys_day, sys_month, and sys_year (which are initialized to noon on 4 July 1776, in case the implementor is careless).

\begin{verbatim}
procedure fix_date_and_time;
    begin sys_time ← 12 + 60; sys_day ← 4; sys_month ← 7; sys_year ← 1776; { self-evident truths }
        time ← sys_time; { minutes since midnight } 
        day ← sys_day; { day of the month } 
        month ← sys_month; { month of the year } 
        year ← sys_year; { Anno Domini } 
        end;
\end{verbatim}

Page B103, replacement for §246 (12/11/20)

246. Of course we had better declare a few more global variables, if the previous routines are going to work.

{ Global variables 13 \equiv
old_setting; 0 .. max_selector;
sys_time, sys_day, sys_month, sys_year: integer;  \{ date and time supplied by external system \}
The enclosing { and } characters of a macro definition are omitted, but an output routine will be enclosed in braces.

routines that should be aborted, but we can sketch the ideas here: For a runaway definition or a runaway balanced text, we will insert a right brace; for a runaway preamble, we will insert a special \cr token and a right brace; and for a runaway argument, we will set long.state to outer.call and insert \par.

function str_toks(b : pool_pointer): pointer; { converts str_pool[b.. pool_ptr – 1] to a token list }

begin continue: get_token; { set cur.cmd, cur_chr, cur_tok }

if cur_tok < left_brace_limit then

help2(“I’m going to ignore the # sign you just used,”)
(“as well as the token that followed it.”); error; goto continue;

help1(“This \read has unbalanced braces.”); align_state ← 1000000; limit ← 0; error;

494. Here is a procedure that ignores text until coming to an \or, \else, or \fi at the current level of \if...\fi nesting. After it has acted, cur.chr will indicate the token that was found, but cur_tok will not be set (because this makes the procedure run faster).
command is being processed. Beware: For historic reasons, this code foolishly conserves a tiny bit of string pool space; but that can confuse the interactive ‘E’ option.

if name = str_ptr – 1 then  { conserve string pool space (but see note above) }

The so-called boundary character of this font; the value of next_char need not lie between bc and ec. If the very last instruction of the lig_kern array has skip_byte = 255, there is a special ligature/kerning program for a boundary character at the left, beginning at location 256 * op_byte +

each portion of a formula is classified as Ord, Op, Bin, Rel, Open, Close, Punct, or Inner, for

begin char_warning(cur_f, qo(cur_c)); math_type(a) ← empty; cur_i ← null_character;

fraction_noad: s ← fraction_noad_size;

cur_loop ← link(cur_loop); link(p) ← new_glue(glue_ptr(cur_loop));
subtype(link(p)) ← tab_skip_code + 1;

stat if tracing_paragraphs > 0 then end_diagnostic(true); tats

stat if tracing_paragraphs > 0 then begin_diagnostic; tats

hn: 0 . . 64; { the number of positions occupied in hc; not always a small_number }

The entry ‘height, §981.’ here and on many later odd-numbered pages should be ‘height = macro, §135.’

to be in the range a ≤ x ≤ b. System error messages should be suppressed when undumping.
loop. (Actually there’s one way to get error messages, via prepare_mag; but that can’t cause
infinite recursion.)

If final_cleanup is bypassed, this program doesn’t bother to close the input files that may still
be open.

Page B533, line 12 of §1333. (11/29/20)

\begin{verbatim}
begin (Finish the extensions 1378); new_line_char ← −1;
\end{verbatim}

Page B534, line 6 of §1335. (11/29/20)

\begin{verbatim}
begin c ← cur_chr; if c ≠ 1 then new_line_char ← −1;
\end{verbatim}

Page B537, line 18 of §1338 becomes two lines (10/05/20)

\begin{verbatim}
begin clear_terminal;
\end{verbatim}

Page B537, lines 11 and 12 from the bottom of §1338 become three lines (04/02/17)

\begin{verbatim}
begin goto breakpoint;
\end{verbatim}

Page B600, the bottom five lines (05/14/19)

they occupy in a typical production system (executable code size for dark blocks, global data
size for light blocks). In this way the chart indicates a total of about 12 × 22 = 264K bytes of
memory, plus 12 × 10 = 120K for the dynamic memory region not shown explicitly. The dynamic
memory is often considerably larger in practice, because it is desirable to accommodate large
macro packages and large pages.

Page Cx, line 4 from the bottom (06/14/20)

\textbf{20 More About Macros} . . . . . . . . . . . . . . . 175

Page C39, lines 10 and 11 become three lines (07/04/20)

that has already been designed. All you’ll see is ‘(io.mf The letter O [79])’ or possibly only ‘(io.mf [79])’, followed by ‘*’. Now the fun starts: You should type

Page C68, lines 9, 28, 35, 36, 38 (11/11/17)

\begin{verbatim}
uniformdeviate -100 -36.1628
z slanted 1/6 (0.16667y+x,y)
(a,b)zscaled(3,4) (-4b+3a,3b+4a)
(a,b)zscaled dir 30 (-0.5b+0.86603a,0.86603b+0.5a)
(a,b)dotprod(3,4) 4b+3a
\end{verbatim}
tom edge of the type. (With plain METAfont’s \texttt{beginchar} each character has a “bounding box” that runs from \((0, h)\) at the upper left and \((w, h)\) at the upper right to \((0, -d)\) and \((w, -d)\) at the lower left and lower right; variable \(d\) represents the depth of the type. The values of \(w, h,\) and \(d\) might change from character to character, since the individual pieces of type need not have the same size in a computer-produced font.)
pixels. (Some typesetting systems use both of these device-dependent amounts to alter their current position on a page, just after typesetting each character. Other systems, like typical dv1 software associated with TeX, assume that chardy = 0 but use chardx pixels.)

Page C113, lines 5–11 from the bottom

\[
s^# := 5pt^#; \quad \text{define\_pixels}(s); \quad \% \text{side of the square} \\
z_1 = (0, 0); \quad z_2 = (s, 0); \quad z_3 = (0, s); \quad z_4 = (s, s); \\
\text{for } k = 1 \text{ upto } 4: \quad z[k + 4] = z[k] + (\frac{2s}{3}, \frac{s}{3}) \text{; \textbf{endfor}} \\
\text{pickup pencircle scaled} .4pt; \quad \text{draw} \quad z_5 \ldots z_8 \ldots z_7 \ldots \text{cycle;} \\
\text{pickup pencircle scaled} 1.6pt; \quad \text{erase draw} \quad z_2 \ldots z_4 \ldots z_3; \\
\text{pickup pencircle scaled} .4pt; \quad \text{draw} \quad z_1 \ldots z_2 \ldots z_4 \ldots z_3 \ldots \text{cycle;} \\
\text{for } k = 1 \text{ upto } 4: \quad \text{draw} \quad z[k] \ldots z[k + 4]; \quad \textbf{endfor}. \\
\]

Page C114, line 7

\[
\text{for } k = 0 \text{ upto } 4: \quad z[k] = \text{center} + (\text{radius}, 0) \text{ rotated}(90 + \frac{360}{5} k); \quad \textbf{endfor} \\
\]

Page C128, lines 13 and 14

changed. Plain METAFONT has a \texttt{tensepath} operation that does this. For example, \texttt{tensepath unitsquare} = (0, 0) -- (1, 0) -- (1, 1) -- (0, 1) -- cycle.

Page C136, lines 18 and 19

only about 0.28 with respect to the initial and final directions; since METAFONT insists that tensions be at least 0.75, this anomalous path could never have arisen if the control

Page C155, line 7

\[
\langle \text{program} \rangle \longrightarrow \langle \text{statement list} \rangle \langle \text{statement} \rangle \textbf{end} \\
\]

Page C160, lines 7–9

might produce a transcript that includes the following diagnostic information:

\[
\text{rotatedaround(EXPR0)(EXPR1)->} \\
\text{shifted-(EXPR0)rotated(EXPR1)shifted(EXPR0)} \\
\]

Page C165, lines 5–7 from the bottom

(i.e., parameters in parentheses), then we name zero or one or two undelimited parameters. Then comes an ‘=’ sign, followed by the replacement text, and \texttt{enddef}. The ‘=’ sign might also be ‘:=’; both mean the same thing.

Page C171, lines 18–20

Chapter 14’s syntax rules for \texttt{(path primary)}, via \texttt{(pair primary)}. A pair expression is not considered to be of type \texttt{path} unless the path interpretation is the only possibility.
Page C176, line 7 from the bottom (07/09/20)
if @#(z_): t_{z} else: f_{z_} \mathrel{:=} x_{z}; endfor

Page C180, line 3 from the bottom (06/24/20)
‘=’ or ‘:=’ following let.

Page C187, line 11 from the bottom (07/12/20)
| substring (pair expression) of (string primary)

Page C189, line 14 (06/13/20)
‘! ’ and followed by ‘,’ , followed by lines of context as in METAFONT’s normal error

Page C200, line 12 from the bottom (08/27/20)
y_1 = y_2 = good.y(5[-d,h]+1.1pt);

Page C202, line 17 from the bottom (06/13/20)
command, and it works only when the penpos angle is 0. If the penpos command is

Page C210, bottom eight lines, and top ten lines of page C211 (07/16/20)

\langle \text{numeric atom} \rangle \rightarrow \langle \text{numeric variable} \rangle \mid \langle \text{numeric argument} \rangle
\mid \langle \text{numeric token primary} \rangle
\mid \langle \text{internal quantity} \rangle
\mid \text{normaldeviate}
\mid (\langle \text{numeric expression} \rangle)
\mid \text{begingroup} \langle \text{statement list} \rangle (\text{numeric expression}) \text{endgroup}
\mid \text{length} (\text{numeric primary}) \mid \text{length} (\text{pair primary})
\mid \text{length} (\text{path primary}) \mid \text{length} (\text{string primary})
\mid \text{ASCII} (\text{string primary}) \mid \text{oct} (\text{string primary}) \mid \text{hex} (\text{string primary})
\mid (\text{pair part}) (\text{pair primary}) \mid (\text{transform part}) (\text{transform primary})
\mid \text{turningnumber} (\text{path primary}) \mid \text{totalweight} (\text{picture primary})
\mid (\text{numeric operator}) (\text{numeric primary})
\mid \text{directiontime} (\text{pair expression}) \text{of} (\text{path primary})
\langle \text{numeric token primary} \rangle \rightarrow \langle \text{numeric token} \rangle / \langle \text{numeric token} \rangle
\mid (\text{numeric token not followed by} ‘/’ \langle \text{numeric token} \rangle’)
\langle \text{numeric primary} \rangle \rightarrow \langle \text{numeric token} \rangle / \langle \text{numeric token} \rangle
\mid (\text{numeric token not followed by} [\langle \text{expression} \rangle, )
\mid \langle \text{numeric atom} \rangle \langle \text{numeric expression} \rangle \langle \text{numeric expression} \rangle \]

Page C214, line 6 becomes two lines (07/17/20)
\langle \text{future pen primary} \rangle \rightarrow \langle \text{future pen argument} \rangle
\mid \text{pencircle}
to see which of its subscripts and suffixes have occurred. For example, if you're

| substring (pair expression) of (string primary) |

Page C217, lines 20–25 (10/07/20)

(program) → (statement list) (non-title statement) end

| (statement list) (non-title statement) dump |

(statement list) → (empty) | (statement) ; (statement list)

(statement) → (empty) | (title)

| (equation) | (assignment) | (declaration) |

| (definition) | (compound) | (command) |

Page C219, line 25 (05/25/20)

This means that the preloaded base you have specified cannot be used, because it is corrupted or was prepared for a different version of METAFONT.

Page C224, lines 7–9 from the bottom (12/21/18)

y₄ᵣ = -0.9848\text{thinn} + 259.00049

x₄ᵣ = -0.08682\text{thinn} + 144

y₄ = -0.4924\text{thinn} + 259.00049

Page C226, lines 9 and 10 (11/01/20)

This means that the preloaded base you have specified cannot be used, because it is corrupted or was prepared for a different version of METAFONT.

Page C228, line 27 (06/19/20)

1.94 endfor

Page C228, line 4 from the bottom (07/12/20)

might want to review now.) You probably also have a proof mode diagram:

Page C234, line 4 of answer 4.6 (07/20/20)

for \(k = 1\) upto 6: \(z[k]' = .2[z[k], z_0] \); endfor

Page C241, line 2 (11/11/17)

\emode=cheapo; input cheaplogo10

Page C242, line 11 of answer 13.7 (07/20/20)

for \(k = 1\) upto 4: \(z[k + 4] = z[k] + (\frac{2}{3}s, \frac{5}{3}s)\); endfor
draw subpath(k, k + 1) of star; cullit;
undraw subpath(k + 2, k + 3) of star withpen eraser; cullit;

def overdraw expr c = begingroup save region;

z_1 = (20, -13); z_2 = (30, -6); z_3 = (20, 1); z_4 = (4, -7);
z_5 = (-12, -13); z_6 = (-24, -4); z_7 = (-15, 6);

path M; M = (origin .. z_1 .. z_2 .. z_3 .. z_4 .. z_5 .. z_6 .. z_7 .. origin .. -z_7 .. -z_6 .. -z_5 .. -z_4 .. -z_3 .. -z_2 .. -z_1 .. cycle)

path z_0 -- z_1 is equivalent to 'z_0 .. controls 1/3[z_0, z_1] and 2/3[z_0, z_1] .. z_1', and the

filldraw bot_serif_edge_4

because it saves a wee bit of time and because ';' often belongs before endfor.
Page C251, replacement for answer 22.1 (07/12/20)

22.1 (a) If and only if \(n\) is an integer between 0 and 255. (b) If and only if \(s\) is a string of length 1.

Page C254, lines 10–13 from the bottom become five lines (06/26/20)

? H
I found no right delimiter to match a left one. So I've put one in, behind the scenes; this may fix the problem.

? 

Page C260, the “line” after line 3 (06/14/20)

\[
\begin{aligned}
\text{font_size} & = (\text{numeric#}); \\
\text{font_slant} & = (\text{numeric#}); \\
\text{font_normal_space} & = (\text{numeric#}); \\
\text{font_normal_stretch} & = (\text{numeric#}); \\
\text{font_normal_shrink} & = (\text{numeric#}); \\
\text{font_x_height} & = (\text{numeric#}); \\
\text{font_quad} & = (\text{numeric#}); \\
\text{font_extra_space} & = (\text{numeric#}); \\
\end{aligned}
\]

Page C261, lines 16 and 17 from the bottom (06/14/20)

\[\{ \text{proofrule} \}, \{ \text{screenrule} \}; \text{makegrid}(\{\text{numerics}\})(\{\text{numerics}\}); \text{proofrulethickness}(\text{numeric#}); \text{proofoffset}(\text{pair}).\]

Page C266, lines 19 and 20 (07/04/20)

You can say either ‘incr x’ or ‘incr (x)’, within an expression; but neither of them are valid statements by themselves.

Page C269, line 11 (01/10/21)

\smode="specmode"; mag=(magnification); input \{font file name\}

Page C277, lines 15–19 (03/06/17)

def openit = openwindow currentwindow from origen % and please correct to (screen_rows,screen_cols) at (-50,300) enddef; % "(-50,300)" too
def showit_ = display currentpicture inwindow currentwindow enddef;
def showit = openit; let showit=showit_; showit enddef; % first time only

Plain METAFONT has several other terse commands similar to ‘openit’ and ‘showit’:

Page C279, line 1 (11/11/17)

blacker:=.1; % make pens a teeny bit blacker
Page C289, line 20 (10/07/20)
if {{(pair x) cand x>(0,0)}}: A else: B fi.

Page C290, line 18 (07/24/20)
save u_; setu_ u; let switch_ = if; if false: enddef.

Page C292, line 10 from the bottom (10/23/20)
be known by saying ‘if known (p − q): p = q else: false fi’; transforms could be handled

Page C293, line 13 and 14 from the bottom (10/27/20)
f(−1) is false! When c → 0, the quantity $a^3 + b^3$ approaches $−∞$ when $c$ is positive, $+∞$ when $c$ is negative. An attempt to ‘solve $f(1,−1)$’ will divide by zero and come

Page C299, bottom four lines of code become five (08/06/20)
primarydef t Bernshtein nn = begingroup save r; r =
begingroup for n=nn downto 2:
  for k=1 upto n-1: u_[[[k]]]:=t[[[[u_[[[k]]]],u_[[[k+1]]]]]]
  endfor endfor u_[[[1]]] endgroup; numeric u_[[[]]]
r endgroup enddef;

Page C305, lines 14–18 (07/08/20)
width_adj#:=-Opt#; % width adjustment for certain characters
serif_fit#:=-Opt#; % extra sidebar near lowercase serifs

Page C305, line 18 (07/08/20)

Page C317, line 21 becomes two lines (11/11/17)
⟨label⟩ → ⟨code label⟩ | ⟨code⟩:: | :
⟨code label⟩ → ⟨code⟩::
Page C318, lines 10–16 from the bottom (11/11/17)

| (code label) (labeled code)  
(deextensible command) \rightarrow extensible (code label) (four codes)  
(four codes) \rightarrow (code), (code), (code), (code)

Notice that a (code label) can appear in a ligtable, charlist, or extensible command. These appearances are mutually exclusive: No code may be used more than once as a label. Thus, for example, a character with a ligature/kerning program cannot also be extensible, nor can it be in a charlist (except as the final item).

Page C333, line 29 (10/25/19)

"if charcode>0:currentpicture:=currentpicture scaled mg;fi;"

Page C333, bottom two lines become one (11/11/17)

if unknown scale: scale := max(1,round(pixels_per_inch/300)); fi

Page C339, line 3 (05/21/20)

ing ‘ß’, ‘æ’, ‘œ’, and ‘ø’) and the uppercase letters (including ‘Æ’, ‘Œ’, and ‘Ø’) are

Page C341, line 14 from the bottom (11/11/17)

prints the \table and the \text; \bigtest gives you the works, plus a mysterious word

Page C345 and following, selected amendments to the index (01/20/21)

(addto command) 118, 220.
bell-shaped distribution 123, 251.
black 270, 332–333.
(code) and (code label) 317.
concatenation of paths 70–71, 123, 127–129, 130, 137, 245, 266.
of strings, 69, 73, 84–85, 187, 278, 286, 312.
*direzione time, 135, 136, 211, 245, 265, 298.
distance, 76, 84, see also length.
dotprod, 68–69, 178, 238, 265.
efficiency, 39, 99, 116, 141, 144, 147, 228, 230, 234, 244, 264, 265, 277, 291, 297, 298.
empty option in for list, 171, 172, 299.
forbidden tokens, 173, 218–219, 286.
*from 191, 220, 252, 277, 312.
Giotto di Bondone, 139.
*independent variables 81–83, 88, 224, 226, 299.
\init, 337, 342.
internal quantities, 54–55, 88, 218, 262, 265–266.
*invwindow 191, 220, 277.
(keep or drop) 118, 220.
labels, 107, 274, 327–328.
*length, 66, 69, 72, 210, 238.


overshoot, 23, 34, 93, 197, 200, 204, 302.

do, 23, 34, 93, 197, 200, 204, 302.

penpos, 36–39, 37, 80, 103, 165, 273, 310.


*p*rotated, 21–22, 25, 27, 44, 68, 73, 107, 114, 117, 141, 213, 238.

rule, 274, 328.

/*scaled, 21–23, 68, 73, 141, 213, 244, 291.

*showstopping*, 211, 219, 227, 230, 262.

string expressions, 69, 187–189, 258, 286.

(suffix list), 171, 236.

sum, of vectors, 9, 68.

test.mf, 311–313.


text arguments, 219, 288–291, 299.

.tfm, 39, 315–321, 333, 335.

*to, 191, 220, 252, 277, 312.

undelimited suffix parameters, 167, 176, 266, 270.

undraw, 113, 118, 120, 242, 271.

unit*square, 116, 123–124, 128, 132, 136, 263.

*unknown, 170, 210.

unknown quantities, nonnumeric, 84–85, 143.


(\texttt{vardef heading}), 165, 178.

*x*scaled, 21–22, 68, 73, 141, 213, 244, 291.

Page Dv, line 16 (01/16/21)

I believe that the final bug in \texttt{METAFONT} was discovered on January

Page Dv, bottom two lines (01/16/21)

corporates all of those changes. I now believe that the final bug was discovered on 03 July 2020 and removed in version 2.71828182. The finder’s fee has converged to $327.68.

Page D2, last line of §2 (01/15/21)

\begin{verbatim}
define banner \equiv 'This is METAFONT, Version 2.71828182' \{ printed when METAFONT starts \}
\end{verbatim}

Page D14, line 1 of §30 (05/05/14)

20. The \texttt{input In} function brings the next line of input from the specified file into available

Page D21, line 8 of §47 (10/11/20)

\begin{verbatim}
g: str_number; \{ the string just created \}
\end{verbatim}

Page D27, lines 3 and 4 of §61 (04/02/17)

is not serious since we assume that this part of the program is system dependent.
Page D28, line 7
var k: 0..23;  { index to current digit; we assume that |n| < 10^{23} }

Page D32, line 2 of §78 becomes two lines
loop begin continue: if interaction ≠ error_stop_mode then return;
clear_for_error_prompt; prompt_input("?\n");

Page D32, line 11 of §79
"E": if file_ptr > 0 then if input_stack[file_ptr].name_field ≥ 256 then

Page D33, line 5 of §80
if file_ptr > 0 then
  if input_stack[file_ptr].name_field ≥ 256 then print("E␣to␣edit␣your␣file.");

Page D37, line 9 of §93
("Try␣to␣insert␣an␣instruction␣for␣me␣(e.g.,␣'I␣show␣x;'),")

Page D82, line 2 from the bottom
define boundary_char = 41  { the boundary character for ligatures }

Page D85, lines 3 and 4 of §194 (and §194 actually moves to page D86)
information, something special is needed. The program here simply assumes that suitable values appear in the global variables sys_time, sys_day, sys_month, and sys_year (which are initialized to noon on 4 July 1776, in case the implementor is careless).

Page D85, the final six lines of §194 (and §194 actually moves to page D86)
procedure fix_date_and_time;
  begin sys_time ← 12 * 60; sys_day ← 4; sys_month ← 7; sys_year ← 1776;  { self-evident truths }
  internal[time] ← sys_time * unity;  { minutes since midnight }
  internal[day] ← sys_day * unity;  { day of the month }
  internal[month] ← sys_month * unity;  { month of the year }
  internal[year] ← sys_year * unity;  { Anno Domini }
  end;

Page D86, replacement for §196
196. Of course we had better declare a few more global variables, if the previous routines are going to work.
  (Global variables +13)+ ≡
old_setting: 0..max_selector;
sys_time, sys_day, sys_month, sys_year: integer;  { date and time supplied by external system }

Page D97, line 2 of §221
the definition of attribute nodes) that it is convenient to let info(p) = 0 stand for ‘[].
but the log \( n \) factor is buried in our implicit restriction on the maximum raster size.) The

\[
\text{for } n \leftarrow 0 \text{ to } n_1 - n_0 - 1 \text{ do env_move}[n] \leftarrow mm0;
\]

direction \((\text{right}_u(p), \text{left}_v(q))\); and there's a line of length \( \geq \delta \) from vertex \( q \) to vertex \( r \),

name points to the eqtb address of the macro being expanded, if the current token list

\[
\text{help2("After } \text{'exitif',<boolean_expr>,}' } \text{I expect to see a ;\semicolon.").}
\]

\{ invokes a user-defined sequence of commands \}

742. Here is a procedure that ignores text until coming to an elseif, else, or fi at the current level of if...fi nesting. After it has acted, \text{cur_mod} will indicate the token that was found.

\[
\text{(A user who tries some shenanigan like 'for . . . let endfor' will be foiled by the get_symbol}
\]

\[
\begin{align*}
\text{begin } \text{wlog(banner)}; \text{slow_print(format_ident)}; \text{print("\_\_\*\}); print_int(sys\_day); print\_char("\_\_\}); months \leftarrow 'JANFEBRMARAPRMAYJUNJULAUGSEPOTNOVDEC'; \\
\text{for } k \leftarrow 3 \ast sys\_month - 2 \text{ to } 3 \ast sys\_month \text{ do wlog(months}[k]); print\_char("\_\_\}); print\_int(sys\_year); print\_char("\_\_\}); print\_two(sys\_time \text{ div } 60); print\_char(":\star\)); print\_two(sys\_time \text{ mod } 60); \\
\end{align*}
\]

command is being processed. Beware: For historic reasons, this code foolishly conserves a tiny bit of string pool space; but that can confuse the interactive ‘E’ option.

\[
\text{if name = str\_ptr - 1 then } \{ \text{conserve string pool space (but see note above) } \}
\]
cur_type = path_type means that cur_exp points to the first node of a path; nobody else points
so-called boundary character of this font; the value of next_char need not lie between bc and ec. If the
very last instruction of the lig_kern array has skip_byte = 255, there is a special ligature/kerning
program for a boundary character at the left, beginning at location 256 * op_byte+
tional halt; no ligature or kerning command is performed.

```
param: array [1..max_font_dimen] of scaled;  { fontdimen parameters }
np: 0..max_font_dimen;  { the largest fontdimen parameter specified so far }
```

```
help1("A colon should follow a headerbyte or fontdimen location."); back_error;
```

to be in the range a ≤ x ≤ b. System error messages should be suppressed when undumping.

If final_cleanup is bypassed, this program doesn’t bother to close the input files that may still
be open.

```
fix_date_and_time; init_randoms(sys_time + sys_day * unity);
```

```
begin clear_terminal;
loop
```

they occupy in a typical production system (executable code size for dark blocks, global data
size for light blocks). In this way the chart indicates a total of about 8 × 22 = 176K bytes of
memory, plus 8 × 15 = 120K for the dynamic memory region not shown explicitly. The dynamic
memory is often considerably larger in practice, because it is desirable to accommodate large
macro packages and large pictures.